

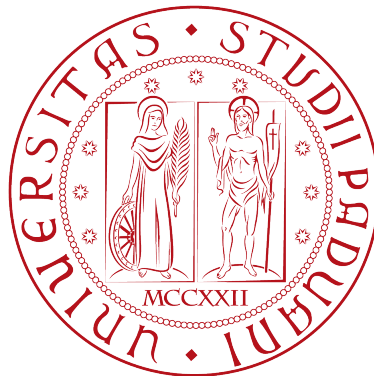
TRASMISSIONE DI CONTENUTI VIDEO IN RETI IP TRAMITE PACCHETTI DI DIMENSIONE AUMENTATA

RELATORE: Leonardo Badia

CORRELATORI: Davide Chiarotto, Marco Mezzavilla

LAUREANDO: Matteo Artuso

A.A. 2011-2012



UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
TESI DI LAUREA TRIENNALE IN INGEGNERIA DELL'INFORMAZIONE

TRASMISSIONE DI CONTENUTI
VIDEO IN RETI IP TRAMITE
PACCHETTI DI DIMENSIONE
AUMENTATA

RELATORE: Leonardo Badia

CORRELATORI: Davide Chiarotto, Marco Mezzavilla

LAUREANDO: *Matteo Artuso*

Padova, 24 Luglio 2012

Indice

Sommario	1
1 Introduzione	2
2 Architettura LTE	5
2.1 Panoramica	5
2.2 Stazione radio base	6
3 Aspetti simulativi	9
3.1 ns-3 e LENA	9
3.2 Modello di canale	11
3.3 Classificazione degli utenti	12
3.4 Implementazione delle simulazioni	13
4 Scenari modellati	15
4.1 Scenari principali	15
4.2 Scenari ibridi	16
5 Studio delle simulazioni	18
5.1 Analisi dei dati	19
5.2 Grafici	21
5.3 Considerazioni sui grafici	23
6 Conclusioni	26
6.1 Conclusioni complessive	26
6.2 Lavori futuri	27
Bibliografia	29

Sommario

La richiesta di infrastrutture e protocolli dedicati per le trasmissioni multimediali in reti radiomobili sempre più performanti è un aspetto sicuramente noto e di profondo interesse sia nel mondo della ricerca che in quello delle soluzioni per le aziende.

Di particolare interesse è la gestione dei file video che implica, in questo ambito, l'ottimizzazione della trasmissione in modo totalmente trasparente all'utente finale per poter offrire un servizio competitivo e performante. Esso, nel tempo, porterà a una radicale riconsiderazione della fruizione dei contenuti video presenti in rete andando a intervenire su quegli aspetti che fino ad ora sono stati un serio handicap alla diffusione su larga scala di contenuti in streaming costante e di prodotti a ciò correlati.

Il lavoro di ricerca di questa tesi, legato al progetto *MEDIEVAL* (*MultimEDIA transport for mobile Video Applications*) [17], è mirato ad un'analisi di simulazioni effettuate con il modulo *LENA*, un simulatore di rete ad eventi discreti sullo stampo di *ns-3* (*network simulator 3*) per pacchetti video in una rete mobile *LTE* (*Long Term Evolution*).

L'idea di base è comparare le variazioni quantitative e dunque qualitative dei parametri di prestazione (ritardo e guadagno) tra l'utilizzare dei pacchetti video di dimensione standard (1500 byte) contro pacchetti considerevolmente più grandi, detti jumbo (9000 byte).

Ciò è mosso dalla semplice e intuitiva idea che, aumentando la dimensione del pacchetto, è possibile ridurre la percentuale riservata alla componente di segnalazione sulla dimensione totale del pacchetto rispetto a quella attualmente prevista nel caso di dimensioni standard.

Capitolo 1

Introduzione

L'idea dei pacchetti di dimensione aumentata, noti in letteratura come *jumbo frames*, viene dall'utilizzo che di essi si fa nelle reti ethernet dove tale tecnica è già consolidata. Si consideri a tal proposito [4].

Nel mutuarla in reti radio (*wireless LAN*) si incontrano indubbiamente delle problematiche non presenti in reti cablate quali la perdita dei pacchetti e la *fairness* [2] che sono state indagate ed approfondite all'interno del progetto MEDIEVAL [17].

Il lavoro di questa tesi si ricollega agli sviluppi fatti e alle soluzioni trovate per le suddette problematiche portando l'utilizzo dei jumbo nelle reti LTE.

L'obiettivo primario dell'intero progetto MEDIEVAL è quello di fornire una serie di miglioramenti sul piano tecnico per l'utilizzo ottimizzato delle risorse video presenti in rete e offrirle all'utente finale garantendo elevati standard qualitativi [17]. Ciò si scontra subito con il substrato sul quale tutto ciò dev'essere realizzato che è la rete LTE (la cui architettura verrà brevemente presentata in seguito nella sezione 2.1).

Vi sono infatti notevoli necessità di ricercare dei compromessi tra quello che gli attuali standard definiti dal *3GPP* (*3rd Generation Partnership Project*) delineano come attuale passo di sviluppo [13] e le prestazioni che si tendono a ricercare a livello simulativo. Considerato ciò si è puntato ad analizzare la struttura del pacchetto IP “standard” [6] e in seguito a ridurre la percentuale di header con ciò intendendo che la dimensione complessiva del pacchetto è stata aumentata a livello simulativo mantenendo invece costante la porzione di header come se si trattasse di un caso standard. Le implicazioni di tale scelta sono che nel pac-

chetto jumbo si ha a disposizione l'equivalente -circa- di 6 pacchetti standard di informazione utile, mantenendo la componente di segnalazione di uno solo di essi. Le prime e immediate considerazioni che si possono fare a riguardo in positivo sono che un minor peso della componente di segnalazione nel pacchetto implica una percentuale superiore di informazione utile disponibile a parità di mole di dati trasmessa, un throughput maggiore e conseguentemente un miglioramento qualitativamente apprezzabile per l'utente finale [11]. Tale miglioramento si traduce quindi in una qualità superiore e in minori tempi di attesa per avere il video disponibile, riducendo i tempi di latenza dovuti al *buffering*¹.

Un immediato aspetto che interferisce con il quadro positivo delineato e quello legato alla robustezza: avere minor segnalazione implica necessariamente pacchetti estremamente più sensibili ai naturali limiti della trasmissione sempre presenti [6], fino al punto -estremo- della perdita del pacchetto che nel caso jumbo comporta la perdita contemporanea di un quantitativo di informazione utile fino a 6 volte superiore del caso standard.

Sono quindi chiare fin da subito due direzioni contrastanti nello sviluppo dell'idea dei pacchetti jumbo; si è perciò pensato di partire con un approccio base al problema in relazione al modello di canale adottato e degli scenari ritenuti descrittivi per una tale indagine. Modello e scenari verranno descritti con maggior dettaglio nei capitoli successivi (Sezione 3.2 e Capitolo 4).

Il fatto che si sia scelto un tal modo di iniziare la fase simulativa è essenzialmente legato alla questione che la tecnica jumbo è molto innovativa in LTE e non si hanno rilevanti riscontri che possa portare i risultati auspicati, di conseguenza qualora non desse nessun evidente vantaggio in scenari basilari, non avrebbe senso pensare di proseguire modellando scenari e modelli di canale più articolati.

Fatte queste doverose premesse sulle assunzioni fatte in fase pre-simulativa si è poi passati a svolgere una campagna di simulazioni al calcolatore tramite il già citato LENA [15] scegliendo scenari ritenuti rilevanti per motivi che verranno precisati nel seguito (Capitolo 4). In seguito all'analisi sui dati raccolti è stato possibile notare come effettivamente ci siano dei vantaggi nell'utilizzo del jumbo, ma non in senso assoluto bensì in alcune tipiche configurazioni scelte il che lascia spazio a sviluppi futuri volti a ripartire dal lavoro preliminare di questa tesi sfruttandone i risultati.

¹Ridurre i tempi di latenza, in generale e non solo nel caso dei video, è uno dei punti di forza di LTE come ricavabile da [9] e anche [10]

Viene ora delineata la struttura della tesi.

Nel capitolo 2 sarà brevemente presentata l'architettura LTE, con particolare attenzione alla porzione che è stata oggetto di modellizzazione a livello simulativo cioè la *Access Network*. Nel capitolo 3 sarà trattato l'aspetto delle simulazioni, mentre nel capitolo 4 ci si dedicherà a delineare con precisione gli scenari prescelti e le motivazioni alla base di tali scelte. Nel capitolo 5 è collocata l'analisi dei dati raccolti al termine della campagna, quali indici di prestazione sono stati scelti e i grafici ad essi relativi. Si riserva infine il capitolo 6 per le conclusioni sulle simulazioni fatte, su ciò che esprimono i grafici, sui limiti individuati e sulle future possibilità di sviluppo.

Capitolo 2

Architettura LTE

2.1 Panoramica

Il 3GPP ha definito, a partire dal 2004, le linee guida per lo sviluppo a lungo termine delle tecnologie cellulari *UMTS* (*Universal Mobile Telecommunications System*) [13]. Nel caso di questo studio si dedicherà particolare attenzione alla parte di accesso al mezzo piuttosto che alla *Core Network* che nel caso di LTE prende il nome di *SAE* (*System Architecture Evolution*) ed è comprensiva di tutti gli aspetti non radio della comunicazione (figura 2.1) come si ricava da [9] e [10].

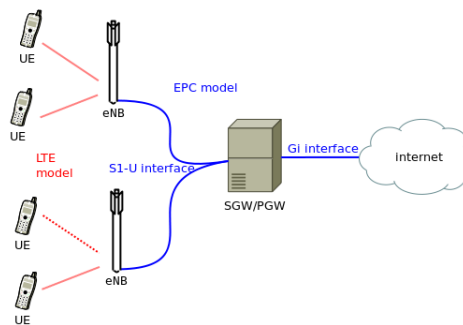


Figura 2.1: Struttura complessiva [10]

La parte terrestre della rete detta *E-UTRAN* (*Evolved UMTS Terrestrial Radio Access Network*) è costituita esclusivamente dalle radio basi *eNodeB* con le quali il terminale utente *UE* (*User Equipment*) comunica direttamente.¹ È in-

¹Nel seguito si farà riferimento a tale *User Equipment* indifferentemente con i termini *utente* e *terminale* essendo tra loro equivalenti in tale contesto.

fatti previsto dallo standard [13] e riportato in [10] che il terminale veda l'intera struttura soprastante semplicemente come un flusso di pacchetti IP (detto *bearer*) caratterizzato da un indice di qualità del servizio (*QoS*). Tuttavia non è l'eNodeB che si occupa di creare e mantenere i bearer, ma ciò è compito dei livelli protocolari superiori, nello specifico i vari tipi di gateway con i quali comunica l'eNodeB, ovvero *MME* (*Mobility Management Entity*) e *S-GW* (*Serving Gateway*) [9] come rappresentato in Figura 2.2.

Una descrizione più dettagliata dell'architettura esula dagli obbiettivi di questo lavoro e si rimanda alla bibliografia (in particolare a [9] e [10]) o a uno dei (numerosi) testi disponibili a riguardo.

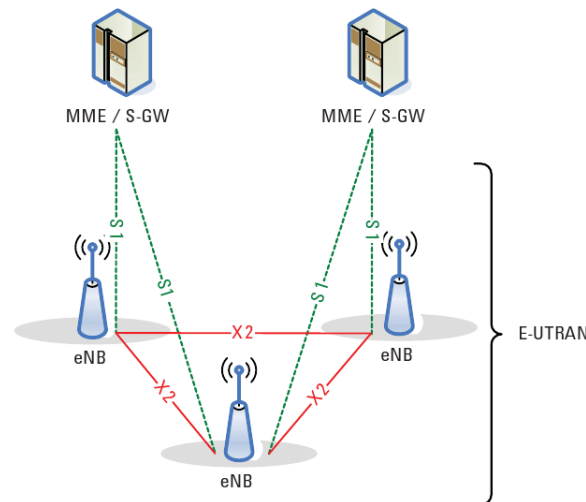


Figura 2.2: Struttura E-UTRAN [10]

2.2 Stazione radio base

L'eNodeB è l'elemento fondamentale per l'allocazione delle risorse [7]: essa riceve il pacchetto dalla *EPC* (*Evolved Packet Core*) in un buffer e si occupa di suddividerlo nei *Resource Block* che sono di fatto le unità base di una griglia tempo/frequenza più complessa (Figura 2.3).

Oltre a questo compito la radio base deve gestire le interferenze tra celle (se è previsto un qualche tipo di coordinamento tra di esse), deve organizzare in modo efficiente lo scheduling e tutto ciò dinamicamente in accordo a [10].

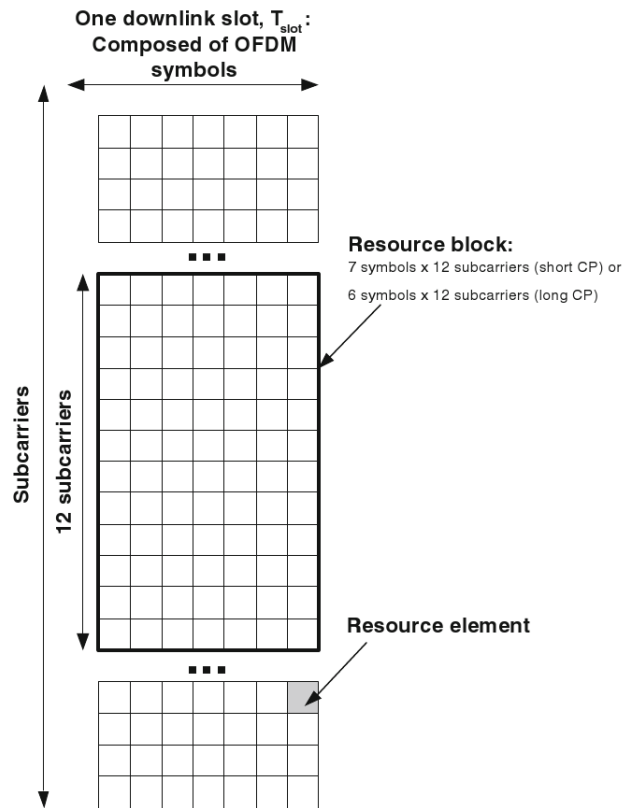


Figura 2.3: Resource Block di eNodeB [10]

Nello specifico: il terminale comunica a eNodeB la condizione del canale che vede sia dal punto di vista del tempo che della frequenza con un indice apposito detto *CQI* (*Channel Quality Indicator*), a tale indice la radio base fa corrispondere un altro valore detto *MCS* (*Modulation and Coding Scheme*) che, come suggerisce il nome, determina quale modulazione debba usare eNodeB sia in uplink che in downlink tra le tre disponibili ossia QPSK, 16QAM, 64QAM. Stabilisce inoltre la codifica di canale tramite il valore normalizzato a 1 della percentuale di bit utili sul totale con l'indice *ECR* (*Effective Code Rate*) come ricavato da [5].

Queste tre modulazioni implicano una diversa mole di risorse allocabili, se infatti il CQI è tale per cui debba essere utilizzata una QPSK si è nel caso della peggior modulazione disponibile in termini di probabilità d'errore sul bit quindi le risorse devono essere allocate in molti più slot e frame rispetto alle altre.

Tornando alla suddetta griglia tempo/frequenza dello scheduler si ha in ascissa il tempo con la durata di un subframe di 1 ms mentre un frame è costituito da 10

2. ARCHITETTURA LTE

di essi (10 ms) e in ordinata si ha la larghezza di banda complessiva disponibile² suddivisa nelle sottoportanti definite dalle modulazioni ortogonali.

Per lo studio in questione il RB (Resource Block) è l'unità di allocazione fondamentale ed esso misura 1 ms in ascissa e 180 kHz in ordinata [13]. Al suo interno una porzione (fissa e tipicamente di 4 bit [10]) è riservata al valore codificato di MCS in base al CQI dell'utente e la rimanente parte è una quantità variabile di bit di informazione utile (e header del pacchetto utilizzato) a seconda della modulazione in downlink stabilitasi con l'utente. Si avrà quindi che più bassa è la probabilità d'errore sul bit (64QAM) maggiore è il numero di bit allocabili in un singolo RB. Ciò avrà conseguenze decisamente molto rilevanti sui risultati delle simulazioni fatte.

Per le strategie di scheduling si ha che in verità in nessuna delle release del 3GPP (a partire dalla prima con LTE che è la release 8 come definito in [13]) viene specificato in modo rigido come debba essere fatto, è perciò un grado di libertà per il progettista della eNodeB che quindi riscontra i problemi di management presentati da [11] riguardo le risorse.

In genere si utilizzano tecniche come il *Round Robin*, probabilmente il più semplice approccio noto in letteratura nella gestione degli utenti in coda. Nel caso in questione si ha che la risorsa da dividere tra i terminali è l'insieme dei RB e la suddivisione viene fatta ciclicamente assegnando una porzione di RB ad ogni utente per ogni ciclo.

Altro tipo di scheduling tipicamente utilizzato è il *proportional fair* che è invece basato su un compromesso tra le prestazioni della rete e la qualità del servizio. A tal proposito si vedano [12] e [3].

²Essa è variabile ed è uno dei punti di forza di LTE come definito in [13] e [10]. La variabilità è tra 1.5 MHz arrivando a un valore massimo di 20 MHz [9]

Capitolo 3

Aspetti simulativi

Andando oltre la descrizione qualitativa fatta sull'architettura della rete d'accesso e del ruolo della radio base, si passa a delineare l'aspetto centrale del lavoro svolto e cioè la componente simulativa.

Il simulatore LENA, sviluppato dal *CTTC (Centre Tecnològic de Telecomunicacions de Catalunya)* [15], è uno dei pochi a questo livello e open-source e perciò è in costante revisione e miglioramento. Esso è appositamente pensato per LTE e implementa la struttura descritta sopra delle radio basi e della gestione delle risorse.

Il vantaggio di utilizzare simulatori di questo tipo è legato alla loro modularità e all'ampia possibilità di riconfigurazione che ciò comporta [18]. Con ciò si intende che, partendo dai moduli standard, è possibile modificare i parametri di default modellando perciò la rete desiderata.

3.1 ns-3 e LENA

La gestione preliminare di ns-3 prevede lo studio dei linguaggi di programmazione in cui è scritto (C^{++} e *python*). In seguito si passa all'analisi della struttura del simulatore, alla sua gestione e alla sua configurazione¹.

Tale studio si è rivelato necessario per acquisire una certa pratica, anche se non puntigliosamente dettagliata, con la struttura stessa dei moduli del simulatore, la loro organizzazione gerarchica e come è consigliato modificarli per le proprie

¹Con l'ausilio degli appositi tutorial e manuale disponibili direttamente dal sito degli sviluppatori [18].

3. ASPETTI SIMULATIVI

ricerche attenendosi agli standard delle licenze open-source di esso.

Nel corso di tale processo si è potuto modellare un certo numero di architetture di rete, nello specifico: una semplice connessione point-to-point tra due nodi, un bus Ethernet² con più nodi connessi, un'infrastruttura mista con una componente radio (wi-fi) e una non radio (bus Ethernet client-server) connesse tra loro, degli scenari introduttivi con utenti wi-fi in movimento casuale nello spazio e infine degli scenari analoghi con palazzi per capire la loro influenza. In tutti questi esempi si è inoltre costruita l'interfaccia tra utenti, si è simulata la rete internet (vista come flussi di pacchetti IP) e l'indirizzamento tramite IPv4. Per i modelli di rete implementati fare riferimento a [18] in particolare al tutorial del simulatore ivi reperibile.

In particolare gli scenari che prevedevano utenti mobili, o comunque un'interfaccia radio a internet, sono stati particolarmente utili per capire le modalità di tracciamento disponibili: ns-3 offre infatti molteplici tecniche di tracking dei pacchetti (alcune pensate esclusivamente per il debugging del codice tipo il *pcap tracking*) ed è possibile impostare quali informazioni far stampare in output a seconda dell'interesse. Ad esempio sarà sicuramente rilevante un indice temporale che stampi il proprio valore ogni volta che il pacchetto viene ricevuto, inviato o viene fatta un'operazione di *acknowledgment* su di esso, come pure l'identificativo dell'utente ad esso relativo che ha svolto tale operazione (indirizzo IP o altro come si vedrà nella Sezione 5.1).

Lo studio di LENA è stato analogo essendo i due simulatori identici in tutto tranne per il modulo di LTE che è molto più sviluppato nel simulatore sviluppato dal CTTC [15].

Con quest'ultimo si è visto come vengono implementati dal simulatore LTE ed EPC, come configurare i vari parametri della rete, come gestire gli output (questione che verrà ripresa in dettaglio nella Sezione 5.1), la modellizzazione del *fading* (vedi Sezione 3.2) e le mappe dei tre tipici scenari (di pedoni, di automobili e urbano), l'utilizzo dei modelli di palazzi in scenari realistici (potendone impostare il tipo, le dimensioni, il materiale, il numero di finestre e così via), le impostazioni possibili sui modelli dell'antenna e infine le mappe delle risorse radio nell'ambiente modellato. Si è inoltre trascurato lo studio della modellizzazione

²Il modello di Ethernet implementato, come si può ricavare dalla documentazione del simulatore, è un sottoinsieme del caso reale che utilizza *CSMA/CD* (*Carrier Sense Multiple Access con Collision Detection*) [18] e anche [6] per le definizioni di CSMA e Ethernet.

della EPC in quanto essa non è rilevante nel lavoro fatto per questa tesi³.

3.2 Modello di canale

Un aspetto sicuramente rilevante nel simulatore è il modello di canale che esso implementa [15] perchè ciò influenza, come già accennato, il CQI e di conseguenza la modulazione del downlink, l'allocazione e la codifica di canale [5]. Per le motivazioni di semplicità della simulazione accennate nell'introduzione e per altre ragioni che diverranno chiare nel seguito si è modellato il canale dimodoché non fosse presente il *multipath*⁴ [8], un fenomeno che si può capire di primo acchito con la figura 3.1.

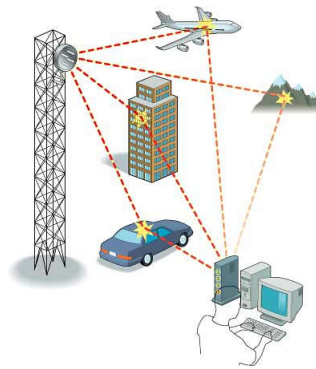


Figura 3.1: Multipath [14]

È un fenomeno di interferenza delle onde inviate dalla radio base che si riflettono su edifici, veicoli, territorio e simili. Così si ha che all'utente non arriva il solo segnale del cammino diretto, ma anche una serie di segnali da cammini riflessi che sono repliche, sfasate nel tempo, dell'originale da cammino diretto. Ciò comporta che il segnale complessivo ricevuto sia una sovrapposizione delle varie componenti andando a influenzare il rapporto segnale-rumore (SNR) spesso in modo non positivo essendo tali segnali se non distruttivi, quantomeno interferenti.

Quando i terminali e le superfici riflettenti sono in movimento diviene necessario

³Anche nel caso di LENA è stata utilizzata la documentazione consistente in un tutorial e in una wiki reperibili a [15].

⁴Tuttavia è possibile utilizzare un canale con multipath in LENA, essendo tale fenomeno modellato come si ricava dalla documentazione presente a [15].

modellare tale fenomeno con un processo gaussiano che incorpora una variabile aleatoria con determinate media e varianza per ogni segnale da cammino riflesso perchè ogni replica ha un'attenuazione diversa rispetto alle altre. Tale fenomeno di distorsione è noto come *multipath fading* [7]. Con tali assunzioni si ha allora che la potenza ricevuta dal terminale (P_{rx}) è attenuata rispetto alla potenza trasmessa (P_{tx}) e l'attenuazione $f(\mathbf{d}, \mathbf{t})$ è una funzione vettoriale e stocastica dei raggi vettore e dei tempi:

$$P_{rx} = P_{tx} f(\mathbf{d}, \mathbf{t}) = P_{tx} \sum_{i=1}^n f(d_i, t_i) \quad (3.1)$$

Con l'utilizzo del multipath fading si ha una modellizzazione del canale più completa, tuttavia nelle simulazioni effettuate si è scelto di non ricorrere a tale modello di canale, ma a uno semplificato nel quale l'attenuazione è funzione della sola distanza tra il terminale ed eNodeB (canale con solo *pathloss*). La motivazione che sta alla base di ciò è ricollegabile al fatto che se tali simulazioni non portassero risultati apprezzabili con un modello di canale così semplice, a maggior ragione non lo farebbero in un canale comprensivo del multipath. Nella stessa ottica si decide di utilizzare un'unica eNodeB, non c'è allora alcuna necessità di gestire le interferenze intercella, la mobilità a bordo cella e la gestione contemporanea di più bearer [9].

3.3 Classificazione degli utenti

Per quanto detto nel paragrafo precedente si ha che il CQI viene determinato esclusivamente dalla distanza, comportando lo stesso MCS, la stessa modulazione in downlink e la stessa codifica di canale per utenti presenti sulla stessa circonferenza centrata nella radio base.

Per le modulazioni valgono le seguenti corrispondenze [1]:

$$1 \leq \text{CQI} \leq 6 \Rightarrow \text{QPSK} \quad (3.2)$$

$$7 \leq \text{CQI} \leq 9 \Rightarrow \text{16QAM} \quad (3.3)$$

$$10 \leq \text{CQI} \leq 15 \Rightarrow \text{64QAM} \quad (3.4)$$

Si definiscono inoltre delle “*classi*” per gli utenti stabilendo che quelli con modulazione **64QAM** (i più vicini alla radio base) appartengano alla **classe A**, quelli con modulazione **16QAM** alla **classe B** e infine quelli con modulazione **QPSK** (i più lontani dalla radio base) alla **classe C**.

3.4 Implementazione delle simulazioni

Si descrive ora il codice utilizzato per le simulazioni.

Lo script si occupa di costruire un sistema LTE+EPC istanziando un’unica eNodeB, ma lasciando modificabile il numero degli utenti da riga di comando. Non è invece direttamente modificabile da riga di comando la dimensione dei pacchetti delle trasmissioni, ma è necessario intervenire su di essa modificando direttamente il codice dello script.

Più nel dettaglio lo script collega l’utente all’eNodeB e inizializza un flusso di pacchetti IP (*bearer*) tra l’utente e un host remoto (collocabile nella EPC e gestito come descritto nel capitolo 2). È importante sottolineare che tale host è unico e che la dimensione totale del file video inviato rimane sempre costante ($\simeq 450$ MByte) al variare del tipo di utente (jumbo o normale), della classe (A, B o C) e dello scenario. Questo per garantire uniformità alle simulazioni e per potersi concentrare esclusivamente sull’analisi comparativa avendo il più basso numero possibile di variabili.

Una volta stabilita la connessione ha inizio la trasmissione del suddetto file video con la simulazione, da parte dei moduli dedicati di LENA, dello scheduling tipico della radio base. La simulazione termina una volta che è stata completata la trasmissione per tutti gli utenti. Ovviamente, per quanto stabilito dalla modellizzazione fatta, essi saranno sicuramente quelli della classe C.

Nel corpo dello script quando si invocano i moduli predefiniti del simulatore è possibile impostare i necessari parametri quali:

- il tempo interarrivo dei pacchetti (pari a 1 ms per omogeneità ai subframe di eNodeB);
- il datarate (100 Gb/s);
- il ritardo intrinseco del canale (10 ms);
- il tipo di protocollo internet (IPv4);

3. ASPETTI SIMULATIVI

- la mobilità degli utenti e della radio base.

Riguardo l'ultimo punto bisogna precisare che necessariamente eNodeB sarà un'entità statica, ma in generale gli utenti potrebbero essere messi in movimento sfruttando i moduli di movimento lungo tre direzioni spaziali in modo pseudo-casuale offerti dal simulatore. Si è tuttavia scelto per questo lavoro di mantenere gli utenti statici, con ciò intendendo che la loro distanza da eNodeB viene fissata una volta per tutte alla creazione dello scenario. Inoltre si avrà pure che utenti della stessa classe saranno tutti sulla stessa circonferenza attorno alla stazione. Tale scelta è giustificata dall'ottica con cui è stata svolta tale simulazione e rientra nell'insieme delle assunzioni fatte nell'introduzione.

Capitolo 4

Scenari modellati

Gli scenari delle simulazioni sono indubbiamente l'aspetto più rilevante da decidere per dimostrare i vantaggi dell'utilizzo del jumbo.

Rimanendo sempre in un'ottica di approccio preliminare alla modellizzazione si ha che una caratteristica comune a tutti gli scenari scelti è l'assenza di edifici tra gli utenti e la stazione radio base. Questo è stato deciso anche per evitare il suddetto fenomeno del multipath, ma pure per modellare una situazione basilare, ma comunque valida per mettere in luce gli aspetti fondamentali della tecnologia in considerazione .

Complessivamente sono state svolte 30 simulazioni ognuna con uno scenario differente. Si può vedere l'insieme delle simulazioni diviso in 6 gruppi, 2 di essi sono i principali e gli altri sono ibridi come verrà poi spiegato. All'interno dello stesso gruppo varia semplicemente il numero di terminali passando da 3 a 15 al massimo divisi uniformemente nelle 3 classi delineate nel capitolo precedente.

4.1 Scenari principali

I 2 gruppi principali di scenari sono quelli nei quali tutti gli utenti presenti utilizzano pacchetti jumbo o normali. Ogni terminale è identificabile da un numero crescente (IMSI) che si ritrova nell'output del tracking dei pacchetti fatto dal simulatore (vedi 5.1). Essi vengono allocati in modo che il primo terzo sia nella classe A, il secondo nella B, l'ultimo nella C a seconda di tale identificativo. Questi due gruppi di scenari permettono, in fase di analisi dei risultati, di confrontare direttamente i due casi estremi che si possono modellare, offrendo inoltre dei va-

lori indicativi dello stato attuale delle cose con gli scenari con tutti gli utenti con pacchetti normali. Questi dati serviranno per un'analisi pesata dell'effettivo guadagno che si riesce a ottenere.

Tali scenari sono inoltre utili per fare un primo e immediato confronto tra le attuali prestazioni della rete e quelle con i pacchetti jumbo nonostante esse siano configurazioni limite e perciò poco probabili in un contesto realistico. Le metriche che si utilizzeranno per analizzare le prestazioni verranno chiarite nel prossimo Capitolo (5) dove verrà svolta l'analisi degli output del tracking.

4.2 Scenari ibridi

Oltre ai due scenari principali si è deciso di modellare anche 4 scenari cosiddetti ibridi con ciò intendendo che comprendono al contempo sia utenti normali che jumbo, allocati di volta in volta in modo diverso nelle tre classi e in percentuali differenti. Prendendo come indice la percentuale di UE jumbo nello scenario si avranno le seguenti configurazioni (con delle sigle aggiunte per questioni di brevità):

- 33% jumbo ed essi nella classe A (quindi se il numero di utenti è 15 quelli con IMSI da 1 a 5 saranno jumbo e nella classe A, quelli da 6 a 10 normali nella B, quelli da 11 a 15 normali nella C), caso **JNN**;
- 66% jumbo ed essi nelle classi A, B, caso **JJN**;
- 33% jumbo ed essi nella classe C (quindi se il numero di utenti è 15 quelli con IMSI da 1 a 5 saranno normali e nella classe A, quelli da 6 a 10 normali nella B, quelli da 11 a 15 jumbo nella C), caso **NNJ**;
- 66% jumbo ed essi nelle classi B, C, caso **NJJ**.

Oltre ai casi elencati si hanno già a disposizione quelli suddetti principali cioè con 0% (**NNN**) e 100% (**JJJ**) di utenti jumbo. Tali scenari ibridi sono stati modellati per vedere se gli eventuali vantaggi dell'utilizzo dei jumbo sono dovuti al fatto che tutti gli utenti usano tali pacchetti aumentati o se invece è possibile avere dei risultati apprezzabili in termini di guadagno anche con una percentuale inferiore di essi (appunto il 33% o il 66%).

C'è stata una particolare supposizione che ha determinato la modellizzazione di

tali scenari ibridi in fase pre-simulativa e cioè che forse solo una determinata classe di utenti avrebbe tratto sostanziale vantaggio dall'utilizzo dei pacchetti jumbo. Tali classi di utenti non possono che essere quelle a modulazione ed ECR migliori cioè quelli della classe A e B, mentre appare difficile che gli utenti della classe C possano trarre vantaggio dalla tecnica per la quantità ridotta di bit che riescono ad allocare in un RB.

Per verificare tale idea i due scenari principali non erano sufficientemente esplicativi perciò si è deciso di modellare anche gli scenari ibridi, per vedere come mutavano i ritardi e i conseguenti guadagni percentuali al variare della frazione di jumbo e della loro collocazione nello scenario.

Capitolo 5

Studio delle simulazioni

L'output di ogni simulazione consiste in un insieme di file di testo con il tracking dei pacchetti nello scenario e un registro di simulazione. Quest'ultimo non è assolutamente rilevante per l'analisi dei risultati essendo un output per il debugging. I file contenenti il tracking sono strutturati a matrice e nelle prossime sezioni verrà delineato il loro utilizzo e l'analisi dei dati fatta tramite uno script in *MATLABTM* [16] appositamente creato per la gestione automatizzata da parte del calcolatore di questa (ingente) mole di dati.

Sempre riguardo alla gestione degli output delle simulazioni si deve sottolineare quale sia stata la tecnica adottata per essi: come già accennato parlando della struttura dello script di LENA si ha che è necessario stabilire prima della compilazione, intervenendo sul codice stesso, che tipo di pacchetti sono utilizzati da ogni classe di utenti, così facendo si fissa il gruppo di simulazioni che si sta effettuando (JNN, NNJ, ecc). Poi il numero di utenti è gestibile da riga di comando e non è necessario intervenire direttamente sul codice, quindi si è utilizzato uno script di tipo *perl* per fare in modo che fosse automatizzata tale operazione infatti una volta scelto il gruppo di scenari lo script si occupa di:

1. impostare il numero di utenti (3 la prima volta);
2. creare una cartella denominata come il numero di utenti all'interno della cartella corrente (che è denominata come il gruppo di simulazioni in corso);
3. salvare in tale sottocartella gli output di tipo .txt dello scenario attuale;
4. terminata la simulazione ne viene subito lanciata un'altra dallo script dopo

aver incrementato di 3 il numero degli utenti e di 1 un contatore (inizializzato a 1);

5. quando il contatore di controllo arriva al valore 5 (quindi è stata conclusa la simulazione con 15 utenti) il ciclo si blocca e tale gruppo di scenari è concluso.

Per svolgere tali simulazioni, si è sfruttato il cluster di calcolo dipartimentale *BLADE* gestendo il trasferimento dei file con il *SFTP* (*Secure File Transfer Protocol*) e un tunnel *SSH* per l'autenticazione sicura.

5.1 Analisi dei dati

Come precedentemente è stato accennato l'output di ogni simulazione è un insieme di file di testo. Tuttavia non tutti sono necessari allo studio in questione, ma solo quelli denominati *DlRlcStats.txt* e *DlMacStats.txt* il cui contenuto è strutturato come segue:

1. file RLC

```
start end    cellId IMSI RNTI LCID nTxPDUs TxBytes    nRxPDUs RxBytes
delay stdDev min      max  PduSize stdDev min max
```

2. file MAC

```
time cellId IMSI frame sframe RNTI mcsTb1 sizeTb1 mcsTb2 sizeTb2
```

Caratteristiche comuni dei 2 file sono l'indice *cellId* che indica un identificativo della cella alla quale fa riferimento il valore corrispondente del tracking (nello studio fatto tale valore è sempre 1 essendo eNodeB unica), l'indice *IMSI* (*International Mobile Subscriber Identity*) e l'indice *RNTI* (*Radio Network Temporary Identifier*) che determinano univocamente gli utenti nella rete e perciò assumono sempre gli stessi valori [15].¹

¹In realtà IMSI è un codice personale spesso sostituito dalla sua versione temporanea per motivi di riservatezza dell'utente, mentre RNTI è un indice temporaneo di per se stesso, tuttavia per le assunzioni semplificative fatte in fase di modellizzazione degli scenari essi andranno a coincidere.

Tra gli altri indici rilevanti dei file per la gestione dei dati si è utilizzato: in RLC *TxBytes* o anche *RxBytes* (essendo uguali per il modello di canale) che come suggerisce la sigla è la dimensione totale in byte del file inviato e ricevuto, mentre in MAC si è utilizzato *time* che è un'indice temporale che viene stampato ogni volta che viene effettuata un'azione su uno dei pacchetti (come era stato accennato nella sezione 3.1).

Si è deciso di utilizzare tali indici perchè di fatto lo studio è quasi esclusivamente incentrato sui tempi di ritardo dell'utente e del sistema utilizzati come indici di prestazione delle due tecniche.

Delineata la struttura degli output e di ogni simulazione e la loro organizzazione nelle cartelle da parte dello script *perl* si passa ora a descrivere lo script in *MATLABTM* utilizzato per la raccolta dei dati rilevanti a seconda degli indici suddetti.

Il programma in *MATLABTM* si occupa di esaminare i vari file MAC presenti. Leggendoli esso aggiorna un indice di tempo crescente per ogni utente per classe e di volta in volta calcola il ritardo medio che alla fine della lettura di ogni file salva in un apposita matrice che sarà disponibile al termine del ciclo di lettura dei 30 file MAC e che conterrà i $3 * 30$ valori dei ritardi medi per ogni classe di utenti per ogni scenario.

Sono ora necessarie alcune precisazioni a riguardo di quanto appena detto: innanzitutto si è detto che lo script per la gestione degli output salva alla fine della lettura un valore medio di ritardo per ogni classe. Ciò significa che, fissato lo scenario e una classe, si hanno tanti valori di ritardo al termine del ciclo di lettura quanti sono gli utenti in tal classe; di essi viene poi calcolato il valore medio e salvato nella già citata matrice. Tale procedimento è corretto dal punto di vista dell'analisi dei dati perchè il valore medio è sicuramente un indice rilevante ai fini dell'analisi comparativa che si dovrà fare, tuttavia in questo caso è pure superfluo perchè per quanto detto in 3.2 non si ha il fenomeno del fading, quindi non ci sono componenti stocastiche di cui valutare l'aspettazione e viene calcolata la media di valori uguali tra loro. Ciò succede per il motivo già descritto che gli utenti collocati alla stessa distanza vedono le stesse condizioni del canale (avendo solo *pathloss*) di conseguenza avranno lo stesso ritardo.

Nonostante questa precisazione si ha comunque che lo script *MATLABTM* calcola un valore medio per motivi di riutilizzabilità del codice qualora si decidesse di

ripetere la campagna simulativa con un modello di canale più realistico comprendente il *fading*.

A tal punto sembra allora inutile l'aver simulato 6 gruppi di scenari 5 volte ciascuno se i ritardi non cambiano, in verità ciò è stato fatto per analizzare come mutano le prestazioni all'aumento della congestione della rete dovuto al crescere del numero dei terminali. Il tutto è infatti visto come un sistema a coda dove il ruolo del servitore è svolto dalla eNodeB con il suo scheduler, mentre la coda sono gli utenti da servire con i pacchetti a loro destinati presenti nel buffer SDU. Un'ultima precisazione sempre a riguardo della raccolta dei dati riguarda il ruolo del file RLC che finora non è stato chiarito: in esso è possibile trovare il valore della dimensione totale del file trasmesso e ricevuto da ogni utente a seconda del proprio identificativo nella rete e ciò torna utile per una prima analisi del *throughput* che era stata fatta vedendolo semplicemente come:

$$\eta = \frac{RxBytes * \frac{1000}{8}}{t_{delay}} \quad \left[\frac{Mbit}{S} \right] \quad (5.1)$$

Tuttavia in fase post-simulativa si è capito che il *throughput* non è un indice eloquente per la comparazione delle due tecniche essendo più rilevante quando il modello di canale usato è più articolato. Per le stesse ragioni di riutilizzabilità del codice suddette e per gli sviluppi futuri lungo le direzioni che verranno poi suggerite si è deciso di includerlo comunque nello script *MATLABTM*.

5.2 Grafici

Per rendere chiari e immediatamente comprensibili i risultati salvati nelle matrici si è deciso, dopo altre tipologie di grafico e plotting meno esplicite, di ricorrere a dei diagrammi a barre accorpati a seconda del gruppo di scenari. Si è inoltre deciso di tenere separati in due diversi grafici un confronto tra gli scenari principali (Figura 5.2) e un altro comprendente pure quelli ibridi (Figura 5.2).

Tale scelta di dividere i risultati in questi due gruppi per i grafici è immediatamente collegata a quanto era stato detto nel Capitolo 4 sulla distinzione tra le due tipologie di scenari simulati: i principali servono per una valutazione immediata e di massima tra la tecnica attuale e uno scenario futuro con solo ed esclusivamente utenti jumbo, mentre quelli ibridi sono per una ricerca più mirata che indaga i vantaggi al variare della collocazione per vedere quanto ciò può influire

5. STUDIO DELLE SIMULAZIONI

sul tempo di servizio complessivo del sistema a coda.

Una nota sui grafici ibridi: si è scelto di non inserire tutte le combinazioni ottenute di utenti, ma solo i casi con 3, 9 e 15 utenti complessivamente, per evitare di rendere i grafici poco chiari per la presenza di troppi dati contemporaneamente. Un'ultima precisazione sugli scenari ibridi riguarda il fatto che nei grafici non sono stati inclusi i risultati relativi ai gruppi etichettati NNJ e NJJ. Ciò sia per i suddetti motivi di chiarezza espositiva, sia -e soprattutto- perchè si è notato dai risultati prima che essi venissero plottati che tali configurazioni risultano poco interessanti e tendono addirittura a far perdere l'ottica d'insieme di quel che si vuole dimostrare con questa tesi.

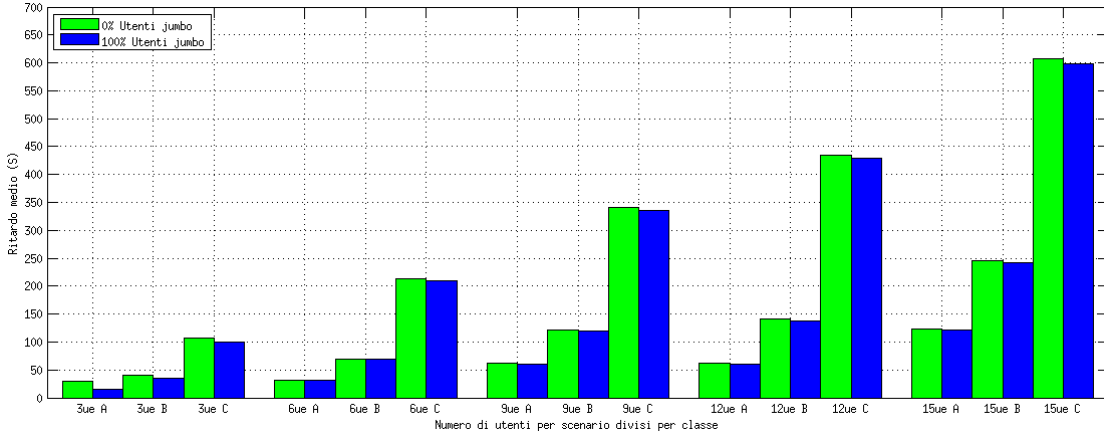


Figura 5.1: Confronto del ritardo tra i casi 0% e 100% utenti jumbo

In aggiunta a questi due grafici che mostrano direttamente uno vicino all'altro i valori dei ritardi nelle diverse condizioni dello scenario modellate si è deciso di inserire un'ulteriore coppia di grafici per far vedere qual è, quantitativamente parlando, il **guadagno** nell'utilizzare la tecnica jumbo in termini di ritardo sulla trasmissione del file completo.

Per essere più precisi a definire tale concetto di guadagno si prendano i due valori di ritardo -detti t_{norm} e t_{jumbo} - corrispondenti a una stessa disposizione degli UE nello scenario, ma riferiti il primo al caso detto NNN e il secondo a uno degli altri presentanti utenti jumbo.

Si definisce allora $g\%$ un guadagno (percentuale) come:

$$g\% = \frac{(t_{norm} - t_{jumbo}) * 100}{t_{norm}} \quad [\%] \quad (5.2)$$

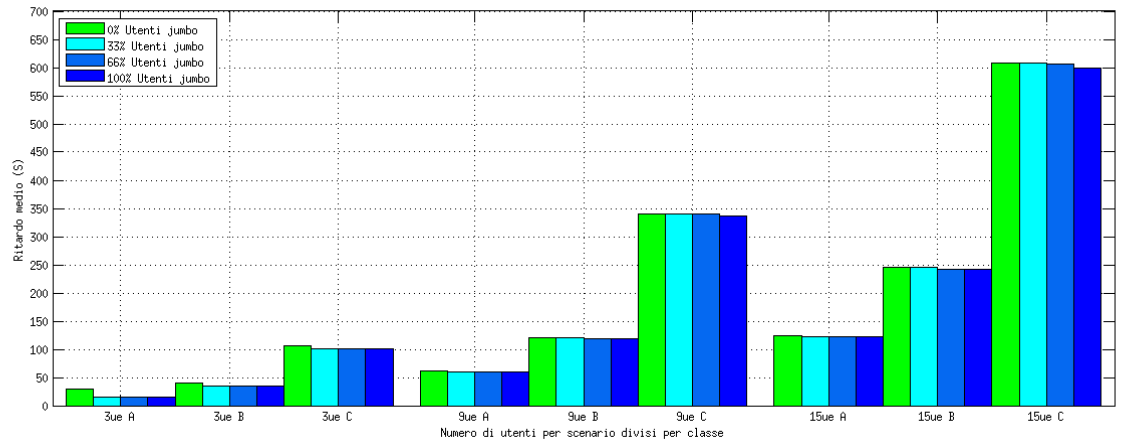


Figura 5.2: Confronto del ritardo tra i casi 0%, 33%, 66% e 100% utenti jumbo

Con questa grandezza appena definita si può descrivere in modo abbastanza immediato e con lo stesso tipo di grafico a barre utilizzato per i valori dei ritardi il guadagno (eventuale) in termini di ritardo, cioè se qualche classe di utenti o addirittura tutti hanno dei vantaggi nell'utilizzo della tecnica jumbo ciò comportando un minor tempo per svuotare la coda di servizio complessivamente o per certe situazioni.

Tuttavia potrebbe anche verificarsi, ed anzi è probabile che sia così, che non ci sia alcun guadagno sensibile dall'utilizzo del jumbo, in particolare per quelle categorie di utenti con modulazione bassa.

Per queste e ulteriori considerazioni sui grafici e per le conclusioni relative ai risultati ottenuti rimandiamo al prossimo capitolo dedicato alla chiusura di questa tesi e alle proposte per ulteriori sviluppi a riguardo.

5.3 Considerazioni sui grafici

Innanzitutto dal grafico in Figura 5.2 si può subito notare che si ha un miglioramento del ritardo confrontando i casi NNN e JJJ anche al variare degli utenti il che sembra essere promettente in relazione a quanto ci si aspettava fin dall'inizio cioè un miglioramento e pure in senso assoluto.

Tuttavia si deve subito puntualizzare che tale miglioramento non è poi così sensibile come ci si potrebbe attendere, soprattutto se si guarda a scenari con un

5. STUDIO DELLE SIMULAZIONI

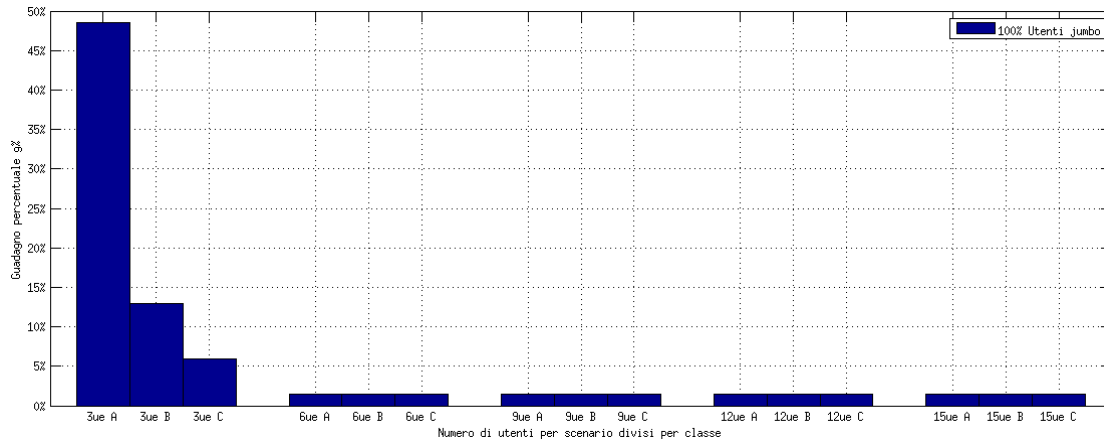


Figura 5.3: Guadagno percentuale $g\%$ in termini di ritardo utilizzando pacchetti jumbo nel caso principale

alto numero di utenti. Un ulteriore conferma di ciò si ha studiando il grafico di Figura 5.2: ivi infatti si nota in modo particolarmente chiaro come il guadagno $g\%$ sia alto e quindi molto interessante se il numero degli utenti è basso.

Altro aspetto da indagare in questi due grafici è che sembra che si abbia qualche miglioramento solo per gli utenti della classe A e che poi le cose peggiorino in modo sensibile e quasi uniformemente all'aumentare degli utenti come pure al cambiamento della classe che, com'è noto, determina il tipo di modulazione in downlink.

Per indagare questo aspetto dei risultati si passa a studiare il grafico della Figura 5.2 che mette a confronto i ritardi medi nei gruppi NNN, JNN, JJN e JJJ.

Qui si nota una conferma di quanto atteso e che era intuibile dal grafico precedente: si ha infatti che l'utilizzo del jumbo comporta dei miglioramenti indubbi, ma che tuttavia sono molto poco sensibili se non nulli nelle configurazioni con 15 utenti. In aggiunta si può notare che la differenza tra il caso totalmente standard e quello jumbo è tanto meno sensibile tanto meno elevata è la percentuale di utenti che utilizzano tale tecnica e ciò si ricollega al fatto che utilizzando pacchetti di dimensione maggiore si può facilmente ricadere in situazioni di coda congestionata e vanno quindi a perdersi quei miglioramenti, seppur esigui, che si riescono a ottenere quando si è nel caso 100% utenti jumbo.

Passando infine a un'analisi del grafico in Figura 5.2 si ha ancora un ottimo guadagno percentuale nel caso di 3 utenti in classe A e tale guadagno è appros-

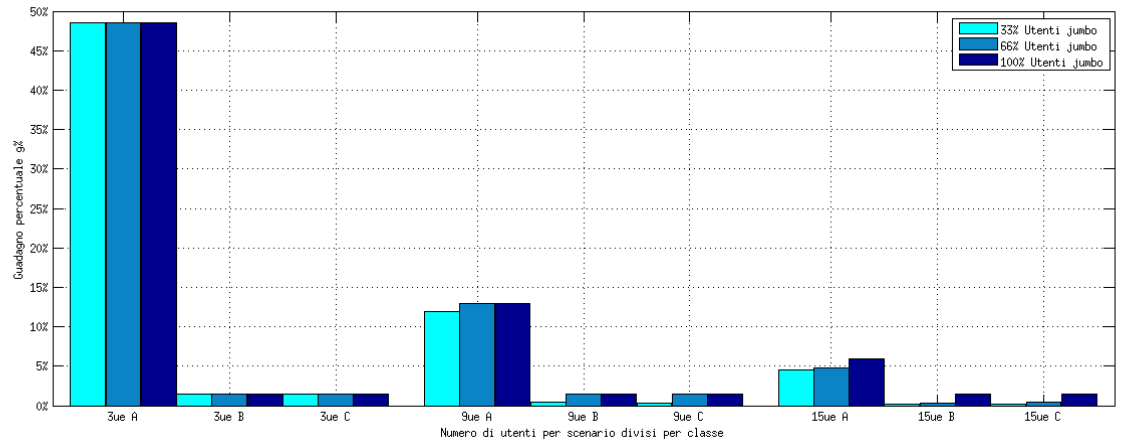


Figura 5.4: Guadagno percentuale $g\%$ in termini di ritardo utilizzando pacchetti jumbo nel caso ibrido

simativamente identico per tutte e tre le configurazioni (JNN, JJN e JJJ) perché gli utenti di classe A usano sempre jumbo. Per questa ragione si è deciso di non includere i risultati dei gruppi NNJ e NJJ, essi infatti collocano i jumbo in classi dove essi non possono sfruttare la modulazione alta e quindi non apportano nessun beneficio nè alla propria trasmissione, nè tantomeno al tempo complessivo di svuotamento della coda.

Sempre da quest'ultimo grafico si nota una sostanziale assenza di guadagno percentuale nelle altre classi di utenti come pure una sostanziale assenza di miglioramento all'aumentare degli utenti tranne, eventualmente, valori di $g\%$ molto bassi e prossimi al 5% per gli utenti di classe A.

Capitolo 6

Conclusioni

Dopo l'esposizione fatta si passa ora a trarre le conclusioni sul lavoro svolto. Si è trattato indubbiamente di una simulazione iniziale data la complessità del problema e tuttavia in base ai risultati ottenuti si è in grado di capire in che direzione ci si dovrà muovere in futuro per sviluppare ulteriormente tale questione e indagarne i *trade-off* più in dettaglio.

È probabilmente questo il contributo personale più rilevante dato al problema con il lavoro di questa tesi e cioè fornire le direzioni degli sviluppi futuri avendo infatti visto quali sono le situazioni nelle quali si hanno vantaggi dall'utilizzo della tecnica jumbo in LTE.

6.1 Conclusioni complessive

In base all'analisi svolta si può allora concludere quanto segue: non si hanno miglioramenti sensibili in senso assoluto nell'utilizzare i pacchetti video IP di dimensione aumentata, ma si hanno dei vantaggi in alcune situazioni come si era previsto fin dall'inizio delle simulazioni.

Tali situazioni corrispondono ad avere utenti jumbo nelle classi a modulazione più alta (64QAM) mentre è di fatto quasi ininfluente che tecnica adottano gli utenti delle altre due classi. Inoltre il numero di utenti per classe e totali è meno influente di quanto si riteneva: di fatto si ha che se gli utenti di classe A usano i pacchetti jumbo liberano prima la coda di trasmissione e ciò si ripercuote pure sugli altri utenti che hanno a disposizione più risorse dello scheduler. Non ci sono invece vantaggi per gli utenti della classe B come invece si era supposto inizialmen-

te.

Tutto ciò ha senso se si pensa che di fatto il pacchetto jumbo arriva siffatto direttamente dalla EPC per come sono state impostate le simulazioni e la eNodeB non fa alcuna decisione in merito a ciò, ma si deve poi occupare di suddividere tale pacchetto di dimensione aumentata, a seconda delle modulazioni, nei RB. Si ha allora che i vantaggi dell'utilizzo dei pacchetti jumbo si hanno se la modulazione corrispondente in downlink permette di allocare più bit possibili nello stesso RB e ciò si ha infatti solo con la 64QAM. Ha insomma senso utilizzare la tecnica jumbo solo per UE tali per cui i loro CQI implicano MCS tali da avere una 64QAM perché a parità di risorse possono allocare un maggior numero di bit nel blocco a differenza di quanto accade con modulazioni più basse.

Ulteriore conclusione è allora che aumentando la dimensione del pacchetto si ha sì un miglioramento in termini di ritardo, ma non sempre dipendendo ciò infatti dallo spazio limitato allocabile nel RB meglio sfruttato dalle modulazioni alte.

Si può pensare allora di non usare più come indici utili ed espressivi il numero di utenti o la dimensione del video da inviare quanto piuttosto il numero di bit allocabili nel RB: tolta da tale spazio una dimensione generalmente fissa riservata ai valori di CQI e MCS rimane una porzione disponibile che va organizzata adattativamente a seconda della modulazione del downlink.

6.2 Lavori futuri

L'idea principale per sfruttare i vantaggi rilevati è di utilizzare un approccio *cross-layer* per la gestione dei pacchetti: invece di far arrivare il pacchetto di dimensione aumentata direttamente dalla EPC a eNodeB, si può fare in modo che sia l'eNodeB a “decidere” se utilizzare un pacchetto jumbo oppure no.

Con ciò intendendo che i pacchetti arrivino di dimensione normale in ogni caso al gateway, in seguito eNodeB analizza la configurazione attuale e se dispone di bit sufficienti nel RB per un pacchetto jumbo lo comunica al gateway, esso accorpa più pacchetti normali del proprio buffer rimuovendo gli header superflui e unendo le componenti di informazione utile e fornisce poi alla radio base il pacchetto jumbo.

Questo tipo di ottimizzazione porterebbe all'indubbio vantaggio di ricorrere al jumbo solo quando tale tecnica risulta vantaggiosa evitando così in toto i pro-

6. CONCLUSIONI

blemi di congestione sopra citati. Altro vantaggio è che non si deve richiedere il pacchetto jumbo alla rete internet che è fuori dalle competenze e dall'ambito di intervento delle aziende di telecomunicazioni. Si ha infine, sempre in positivo, l'ausilio di un feedback costante tra le due entità della rete in comunicazione.

D'altro canto un tale approccio comporta dei ritardi dovuti alle necessarie comunicazioni continue tra eNodeB e S-GW altrimenti assenti ed è necessaria un'ottimizzazione dimodoché esse avvengano in finestre temporali tali per cui non venga affetto troppo pesantemente il tempo complessivo di gestione della trasmissione. La questione a riguardo è particolarmente problematica se gli utenti sono in movimento, a tal punto infatti CQI e MCS mutano e quindi l'eNodeB deve implementare un algoritmo per fare delle previsioni statistiche sui loro futuri movimenti, a patto che la loro velocità rimanga nei limiti operativi del suddetto algoritmo.

Ulteriore idea da sviluppare è quella di modellare scenari più realistici sia per il modello di canale sia per il tipo di scenario includendo quindi edifici, mobilità degli utenti, celle multiple, gestione delle interferenze e la trasmissione ai bordi delle stesse son tutte le complicazioni che ciò comporta per vedere se i miglioramenti individuati permangono oppure no.

Un aspetto sicuramente rilevante è che, come accennato nell'introduzione, l'utilizzo della tecnica jumbo è legata a scenari avveniristici e permette ottimi vantaggi quando si può garantire una modulazione molto alta e forse potrebbe essere più interessante studiare il tutto con una larghezza di banda totale non di 20 MHz come previsto dagli attuali standard di LTE, ma con una banda a ben 100 MHz prevista invece per *LTE-Advanced* previsto come fase di sviluppo successiva nell'arco dei prossimi 3 anni.

In conclusione si può quindi affermare che le possibilità di miglioramento esistono e possono essere interessanti, ma che necessitano di ulteriori indagini di tipo simulativo e implementativo dello standard per futuri sviluppi.

Bibliografia

- [1] L. Anchora, L. Badia, M. Mezzavilla, M. Zorzi, Simulation Models for the Performance Evaluation of Spectrum Sharing Techniques in OFDMA Networks, *Proceeding MSWiM '11 Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, 2011.
- [2] A.P. Iyer, Deshpande, G., Rozner, E., Bhartia, A., Lili Qiu, Fast Resilient Jumbo frames in wireless LANs, *17th International Workshop on Quality of Service (IWQoS)*, 2009.
- [3] R. Kwan, Proportional Fair Multiuser Scheduling in LTE, *Signal Processing Letters, IEEE*, 2009.
- [4] S. Narayan, Impact on network performance of jumbo-frames on IPv4/IPv6 network infrastructure: An empirical test-bed analysis, *IEEE 4th International Conference on Internet Multimedia Services Architecture and Application(IMSAA)*, 2010.
- [5] *Evolved Universal Terrestrial Radio Access (E-UTRA), Physical Layer Procedures*, 3GPP TS 36.213, Release 8, 2008.
- [6] N. Benvenuto, M. Zorzi, *Principles of Communications, Networks & Systems*, Wiley, London, 2011.
- [7] J.D. Gibson, *The Mobile Communications Handbook*, IEEE press, Boca Raton, 1996.
- [8] A. Goldsmith, *Wireless Communications*, Cambridge University press, Cambridge, 2005.

BIBLIOGRAFIA

- [9] H. Holma, A. Toskala, *Lte for UMTS OFDMA and SC-FDMA based Radio Access*, Wiley, Chichester, 2009.
- [10] A. Hamza, *Long Term Evolution (LTE) - A Tutorial*, 2009 [slides]
- [11] S. Kim, J. Zander, *Radio Resource Management for Wireless Network*, Artech House, Boston, 2001.
- [12] R. Yves, *Introduction to Scheduling*, CRC, Boca Raton, 2010.
- [13] 3GPP A Global Initiative, <http://www.3gpp.org>
- [14] IEEE Spectrum, <http://spectrum.ieee.org>
- [15] LENA LTE-EPC Network Simulator, <http://iptechwiki.cttc.es>
- [16] MATLABTM, <http://www.mathworks.it/products/matlab>
- [17] MEDIEVAL MultiMEDia transport for mobile Video Applications, <http://www.ict-medieval.eu>
- [18] ns-3 Network Simulator 3, <http://www.nsnam.org>